



Universitat
Autònoma
de Barcelona



(5106-5:CAMPUS VIRTUAL UAB)

Memòria del Projecte Fi de Carrera
d'Enginyeria en Informàtica
realitzat per
Josep Bacardit Sáez
i dirigit per
Ramon Grau Sala
Bellaterra 16 de setembre de 2013

El sotasignat, Ramon Grau Sala

Professor/a de l'Escola Tècnica Superior d'Enginyeria de la UAB,

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per en Josep Bacardit Sáez

I per tal que consti firma la present.

Signat: Ramon Grau Sala

Bellaterra, 16 de setembre de 2013

Índex

1. INTRODUCCIÓ	4
1.1 MOTIVACIÓ.....	4
1.2 OBJECTIU DEL PROJECTE.....	6
2. ESTUDI DE LA VIABILITAT DEL PROJECTE.....	7
2.1 ESTAT DE L'ART DEL PROJECTE	7
2.2 TECNOLOGIA A EMPRAR.....	8
2.2.1 <i>El sistema operatiu Android</i>	8
2.2.2 <i>MAMP i PhpMyAdmin</i>	11
2.2.3 <i>Google Cloud Messaging</i>	12
2.2.4 <i>Entorn de desenvolupament</i>	13
3. TASQUES I PLANIFICACIÓ DEL PROJECTE	14
3.1 DISSENY DE L'APLICACIÓ	16
3.2 DISSENY DE LA BASE DE DADES	18
3.3 IMPLEMENTACIÓ DE LA INTERFÍCIE.....	20
3.3.1 <i>El Manifest Android</i>	20
3.3.2 <i>Organització del codi</i>	22
3.3.3 <i>Flux del programa</i>	23
3.3.4 <i>Estructura de dades</i>	27
3.4 ACCESSOS A LA BASE DE DADES	28
3.4.1 <i>La part de l'aplicació</i>	29
3.4.2 <i>La part del servidor</i>	31
3.5 NOTIFICACIONS	33
3.5.1 <i>Implementació GCM Client</i>	34
3.5.2 <i>Implementació GCM Servidor</i>	35
3.6 PROVES.....	36
3.6.1 <i>Terminals</i>	37
3.6.2 <i>Android Log</i>	39
4. CONCLUSIONS.....	40
4.1 OBJECTIUS ASSOLITS I MILLORES	40
4.2 A ON ES VOL ARRIBAR?.....	42
5. BIBLIOGRAFIA	44

1. Introducció

En aquest part es farà un breu resum de les motivacions que m'han dut a portar a terme aquest projecte i dels objectius del mateix.

1.1 Motivació

Pels voltants de l'any 2008, l'època en que van aparèixer els primers smartphones amb sistema operatiu Android vaig tenir la necessitat d'una aplicació per a mòbils del Campus Virtual. Un any després va ser quan em vaig interessar per la programació en Android. Vaig començar a programar coses senzilles però a la vegada molt útils pel meu aprenentatge en aquesta plataforma.

L'aplicatiu del Campus Virtual via web és una eina molt bona, però té un defecte, per poder-te assabentar dels avisos, notes o qualsevol material nou, simplement has d'entrar amb el teu NIU i contrasenya a través d'un navegador. Però que passa si no hi entres? Doncs, no et podràs assabentar de res. Per tant, una de les avantatges que té una aplicació per a mòbils són les notificacions. Moltes vegades ens ha passat als estudiants que un dia qualsevol de classe, has de matinar i estar a primera hora a classe, i quan arribes a l'aula et trobes un paper que ha posat el professor que es suspèn la classe per falta d'assistència del professor i al cap d'una estona, entres al Campus Virtual i veus que hi ha una avís dient que no hi havia classe. També ens ha passat als alumnes amb les notes dels parcials i dels exàmens finals que un cop realitzat l'examen has d'estar mirant cada dia el

Campus Virtual per saber si han penjat les notes. Més d'un cop han trigat fins a 3 setmanes o més per penjar unes notes. Un altre exemple de preocupació dels alumnes es quan hem d'anar a una aula per fer una classe i no sabem a quina aula hem d'estar, per això es pot veure els calendaris penjats al passadís de secretaria o entrar un moment amb l'ordinador al Campus Virtual i veure la informació de l'assignatura. Però tot això es molt incòmode, que fàcil i ràpid seria si tinguéssim una aplicació al mòbil que ens cobreixi totes aquestes mancances. Si un dia no hi ha classe, el professor pot enviar una notificació a cada alumne i automàticament tothom la rep al instant, que fas un examen i no vols perdre el temps mirant si estan les notes penjades, doncs un cop penjades les notes et pots assabentar amb una notificació, si no saps a quina aula has d'anar per fer classe i resulta que estàs a l'altra punta de la facultat, doncs, entres amb el mòbil a l'aplicació i veus la informació de l'assignatura. A més per la gent que viatgi amb transport públic, te la possibilitat de repassar els apunts descarregant-los a través de l'aplicació d'una manera molt directa i ràpida sense haver de treure fulls ni remoure la cartera.

Aquestes son algunes de les raons per les quals sempre he desitjat una aplicació del Campus Virtual. Sempre he estat esperant que algun dia algú pogués fer aquesta aplicació, que et permetés realitzar les mateixes tasques que via web. Com que fins a ara no s'ha fet realitat aquest somni, he decidit fer-ho jo, encara que segurament no la pugui gaudir massa, però pels nous alumnes pot ser molt útil i com que la programació per Android és una de les meves passions, aquest

projecte era una bona oportunitat per fer una cosa útil i a la vegada que m'agradés.

1.2 Objectiu del projecte

L'objectiu principal del projecte es poder facilitar l'accés al Campus Virtual pels alumnes que disposin d'un dispositiu Android. La idea de l'aplicació sempre ha estat que es puguin fer les mateixes coses que es fan via web i que, a més a més, sigui d'un disseny senzill e intuïtiu. Amb l'aplicació es podran veure els apunts de cada assignatura, les notes de cada parcial, els avisos i notícies de cada assignatura. Tindrem a l'abast la informació de cada assignatura. Per cada assignatura sabrem en quines aules s'impartiran les classes, quins professors hi hauran amb els seus mails i despatxos corresponents.

L'alumne només haurà de identificar-se el primer cop que obri l'aplicació i un cop identificat ho tindrà tot al seu abast. A més té el afegit de poder-te comunicar amb els teus companys de classe. D'aquesta manera també serveix com una eina de comunicació entre els alumnes.

En definitiva, això no només ha de ser un projecte final de carrera, sinó una eina pels futurs estudiants de la UAB.

2. Estudi de la viabilitat del projecte

En aquesta part farem una breu comparació amb les aplicacions que hi han al mercat que tenen algun punt en comú amb aquest projecte i comentarem la tecnologia requerida per al desenvolupament d'aquest projecte.

2.1 Estat de l'art del projecte

En referència a la UAB s'han trobat dues aplicacions que tenen l'objectiu de ser una eina per ajudar a l'alumne. En primer lloc, *"CV UAB for Android"* és una aplicació no oficial que ens permet veure les novetats del Campus Virtual. El problema que té aquesta aplicació es que no et notifica quan hi ha alguna novetat i quan veus la novetat ja no la pots tornar a visualitzar en un altre moment perquè desapareix. I en segon lloc, *"UAB Consulta de Qualificacions"* només ens permet visualitzar les notes finals de cada assignatura. A part que només te aquesta funcionalitat, no ens notifica quan hi ha alguna nova nota.

Fora de la UAB hi han aplicacions que estan relacionades amb el tema del projecte. Hi han un parell que es poden destacar.

En primer lloc, *"Universitarios CLM"* és una aplicació del Campus Virtual de la Universidad de Castilla la Mancha. Amb aquesta aplicació pots accedir a tota la informació que necessita un alumne. Es poden consultar apunts, notes, notícies, etc. Li manca la comunicació entre alumnes.

En segon lloc, “Mi UOC Móvil” és una aplicació per la UOC que et permet consultar la teva bústia, agenda, notes i notícies. Li manca el poder veure els apunts de les assignatures i la comunicació entre alumnes.

I per últim, la resta d’aplicacions són semblants però no tan complertes com les anteriors.

Així doncs, aquest projecte pot ser una eina molt completa que facilitarà l’estudi i permetrà al alumne tenir un seguiment més al dia de les assignatures que estigui cursant.

2.2 Tecnologia a emprar

En aquest apartat definirem totes les tecnologies que s’han emprat per el desenvolupament del projecte.

2.2.1 EL SISTEMA OPERATIU ANDROID

Android és un sistema operatiu basat en Linux dissenyat principalment per a telèfons i tablets. Encara que últimament se està expandint molt i ja es pot trobar a reproductors MP3, televisors, rentadores, microones, etc.

L’any 2003 *Andy Rubin, Rich Miner, Nick Sears i Chris White* van crear Android Inc. Un projecte per crear alguna cosa nova i diferent encara que no crec que hagin pensat en l’impacte que ha tingut a dia d’avui Android.

L'any 2005 Google va comprar Android Inc. En aquesta època Google guarda un secretisme i es rumorejava que estaven fabricant un nou sistema operatiu per la telefonia mòbil.

No va ser fins l'any 2007 es va fundar la *Open Alliance Handset* i quan es va fer l'anunci oficial d'Android. Al 2008 apareix el primer telèfon mòbil amb sistema operatiu Android i a partir d'aquest moment, Android no ha parat de créixer i ara mateix és el sistema operatiu més utilitzat en telefonia mòbil.

Hi han més sistemes operatius per a telèfons mòbils diferents a Android. Els més populars són IOS fundat *Apple*, Symbian desenvolupat majoritàriament per *Nokia* i Windows Phone creat per *Microsoft*.

L'elecció d'Android com a sistema operatiu pel projecte ha estat simplement per ser el sistema operatiu més utilitzat i per la seva facilitat de programació, apart de que es pot trobar molta informació sobre Android. IOS era una possible opció també, però per poder programar per IOS es necessari posseir d'un MAC OS i als inicis del projecte no en tenia accés a cap.

En la actualitat hi han 18 versions de la API d'Android. A la següent taula podem veure les diferents versions de la API d'Android amb els seus noms.

Platform Version	API Level	VERSION_CODE
Android 4.3	18	JELLY_BEAN_MR2
Android 4.2, 4.2.2	17	JELLY_BEAN_MR1
Android 4.1, 4.1.1	16	JELLY_BEAN
Android 4.0.3, 4.0.4	15	ICE_CREAM_SANDWICH_MR1
Android 4.0, 4.0.1, 4.0.2	14	ICE_CREAM_SANDWICH
Android 3.2	13	HONEYCOMB_MR2
Android 3.1.x	12	HONEYCOMB_MR1
Android 3.0.x	11	HONEYCOMB
Android 2.3.4 Android 2.3.3	10	GINGERBREAD_MR1
Android 2.3.2 Android 2.3.1 Android 2.3	9	GINGERBREAD
Android 2.2.x	8	FROYO
Android 2.1.x	7	ECLAIR_MR1
Android 2.0.1	6	ECLAIR_0_1
Android 2.0	5	ECLAIR
Android 1.6	4	DONUT
Android 1.5	3	CUPCAKE
Android 1.1	2	BASE_1_1
Android 1.0	1	BASE

Taula 1 - Versió de la API d'Android

En la següent imatge podrem veure un gràfic amb el percentatge d'ús de cada versió de Android en el moment d'iniciar-se aquest projecte.

Version	Codename	API	Distribution
1.6	Donut	4	0.2%
2.1	Eclair	7	2.4%
2.2	Froyo	8	9.0%
2.3 - 2.3.2	Gingerbread	9	0.2%
2.3.3 - 2.3.7		10	47.4%
3.1	Honeycomb	12	0.4%
3.2		13	1.1%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	29.1%
4.1	Jelly Bean	16	9.0%
4.2		17	1.2%

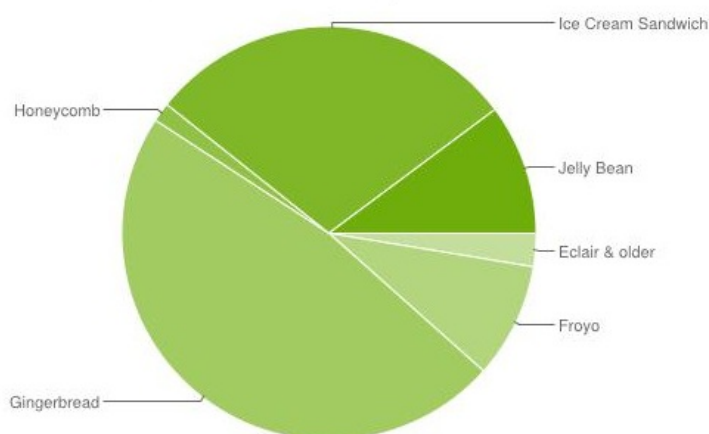


Figura 1 - Gràfic de versions d'Android

L'elecció de la versió de la API d'Android utilitzada per aquest projecte a estat Gingerbread, el nivell 10, al ser la més utilitzada fins al moment i per ser suficientment completa per poder desenvolupar totes les parts del projecte.

2.2.2 MAMP I PHPMyADMIN

Gairebé totes les aplicacions que han de llegir o moure dades utilitzen una base de dades. A causa de no poder aconseguir un servidor extern, vaig haver de fer ús d'un servidor local.

Un cop iniciat el projecte vaig poder posseir un MAC i per tant, la meua elecció com a software per poder fer els accessos a la base de dades va ser MAMP (“Mac Os X”, “Apache”, “MySQL”, “PHP, Pearl o Python”).

D'altra banda, per poder gestionar la base de dades vaig utilitzar la eina PhpMyAdmin. Aquesta eina té una interfície que et permet gestionar una base de dades. Vaig escollir aquesta eina per gestionar la base de dades perquè ja la coneixia i ja havia fet ús en alguna ocasió i considerava que seria la eina més fàcil d'utilitzar pel meu propòsit.

2.2.3 GOOGLE CLOUD MESSAGING

Tot el món coneix les notificacions Push com el típic missatge que ens surt a la pantalla d'un telèfon mòbil quan rebem un Whatsapp o un correu electrònic. Aquest tipus de missatge, des del punt de vista d'Android, es monten amb el servei de Google Cloud Messaging. La manera per treballar aquest servei comença des de l'aplicació enviant un missatge al servidor, seguidament el servidor envia a Google el missatge juntament amb tots els identificadors dels terminals que rebran aquest missatge i a continuació Google envia el missatge a tots els terminals indicats.

2.2.4 ENTORN DE DESENVOLUPAMENT

Pel desenvolupament de la programació s'ha utilitzat les següents eines:

- **Eclipse IDE i Java Standard Development Kit (SDK):**
Eclipse es un entorn de desenvolupament integrat que ens permet programar en diferents llenguatges. Un dels llenguatges que permet es JAVA que és l'utilitzat en el nostre cas. Per poder programar en JAVA també es necessari la instal·lació del SDK de JAVA, que és un kit de desenvolupament pel llenguatge JAVA.
- **Android SDK:** aquesta eina ens proporciona els drivers del nostre terminal Android i ens permet connectar-lo per poder fer les nostres simulacions. En cas de no tenir un terminal Android es pot fer ús d'un emulador.
- **ADT Eclipse Plugin:** és un afegit d'Eclipse i ens permet descarregar les versions d'Android que desitgem, crear emuladors amb diferents característiques, veure el Log d'informació i errors del terminal o emulador, etc.
- **Netbeans:** es un entorn de programació principalment per la realització de pàgines web. Amb aquesta eina s'han pogut realitzar les connexions a la base de dades amb el llenguatge PHP i HTML.

3. Tasques i planificació del projecte

El projecte s'ha dividit en diferents tasques de disseny, implementació i proves. A continuació definim cadascuna d'aquestes tasques:

- **Disseny de l'aplicació:** Aquesta tasca consisteix en com se structuren cadascuna de les pantalles, com estarà organitzada la informació, el botons que apareixen, etc.
- **Disseny de la base de dades:** Aquesta tasca consisteix en com s'estructura la base de dades amb les diferents taules necessàries per garantir la màxima informació per l'aplicació amb els seus camps corresponents.
- **Implementació de l'interfície:** Aquesta tasca consisteix en la organització dels tipus de dades que utilitza l'aplicació juntament amb la implementació del disseny de l'aplicació.
- **Accessos a la base de dades:** Aquesta tasca consisteix en la implementació en PHP i JSON dels diferents accessos a la base de dades juntament amb la implementació a la aplicació per recollir les dades que ens retornen les execucions dels PHP.
- **Notificacions:** Aquesta tasca consisteix en la implementació de les notificacions. Aquestes notificacions inclouen els missatges dels avisos, notícies i la comunicació entre alumnes.
- **Proves:** Aquesta tasca consisteix en la detecció de errors, la millora de noves funcions i la que garantitza el bon funcionament de l'aplicació. Aquesta tasca es combina amb la resta de tasques, ja que, en tot moment es va revisant el projecte per optimitzar el temps.

La durada de cada una de les tasques es veu reflexada a la següent taula.

Tasques	Data d'inici	Duració(dies)	Data final
Disseny Aplicació	01/02/13	14	15/02/13
Disseny Base de Dades	15/02/13	7	22/02/13
Implementació interfície	01/03/13	60	30/04/13
Accesos Base de Dades	30/04/13	30	30/05/13
Notificacions	30/05/13	90	28/08/13
Proves	14/02/13	200	02/09/13

Taula 2 - Planificació de les tasques

Com es pot observar la data prevista per la finalització es el 2 de setembre del 2013, superant el primer termini d'entrega del projecte. Si ens fixem, la tasca de proves comença gairebé al inici del projecte i dura fins el final del mateix. Contínuament s'han anat fent proves per garantir el bon funcionament de cada una de les tasques.

En el següent gràfic veiem el Diagrama de Gantt de la planificació de les tasques.

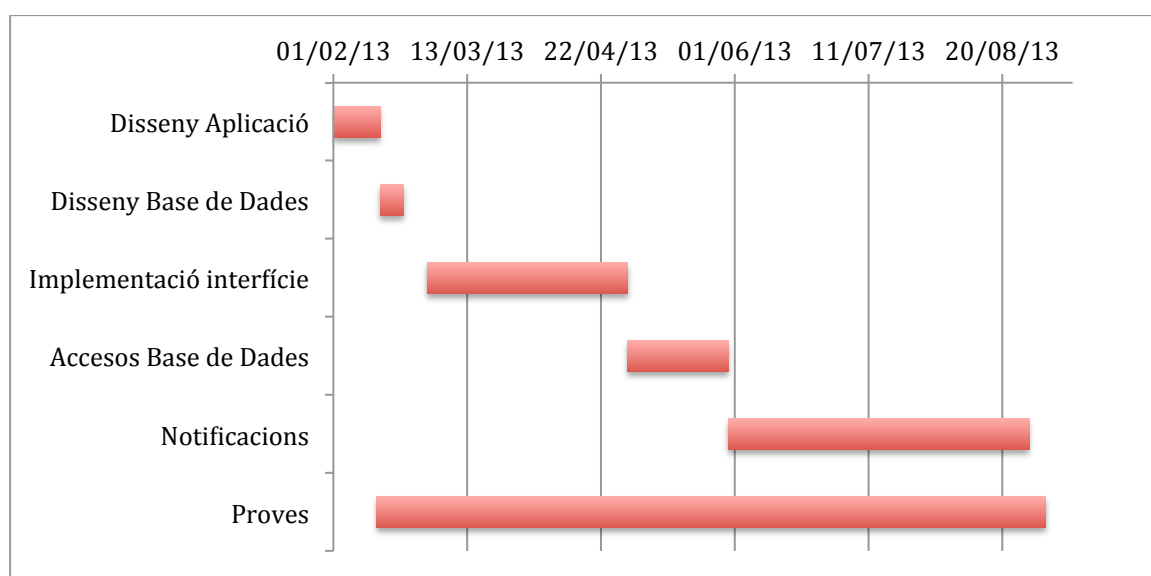


Figura 2 - Diagrama de Gantt

3.1 Disseny de l'aplicació

Abans de començar a programar s'ha de tenir ben clar que es vol fer i com s'ha de fer. Per tant, el primer que s'ha de fer es un disseny de l'aplicació a nivell visual. Amb el disseny de l'aplicació s'ha intentat aconseguir que sigui de fàcil ús i que contingui en tot moment la informació necessària. Els errors més típics en dissenys per aquest tipus d'aplicacions són:

1. Saturar d'informació al usuari.
2. Abusar de tecnicismes que l'usuari no entén.
3. Donar prioritat a la forma sobre el contingut, i sacrificar part de la informació a la accessibilitat.

Encara que està enfocat al contingut, s'ha intentat fer un disseny senzill però a la vegada elegant, intentant combinar els colors del "logo" de la UAB amb els colors del Campus Virtual via web.

Per la realització del disseny s'ha utilitzat l'eina Photoshop amb unes plantilles especials per dissenyar aplicacions d'Android. A més s'han utilitzat un seguit d'icones especials per aplicacions i s'ha intentat seleccionar els icones més clarividents amb la seva funcionalitat.

S'ha de tenir en compte que el sistema operatiu Android es suportat per molts tipus de terminals mòbils. Per tant, per poder fer un disseny preparat per qualsevol gama de terminals s'han de poder fer diferents mides per imatges, icones i qualsevol element visual. La

gran demanda de telèfons amb aquest sistema operatiu ha portat a les companyies a dissenyar aparells per tal de poder arribar a qualsevol usuari i per tant, s'han agut de fer una classificació general en 4 tipus en quant als preus:

- Gama baixa: ZTE Blade, Huawei Ivy, Huawei Sonic, etc.
- Gama mitjana: Samsung Galaxy SCL, HTC Desire, etc.
- Gama alta: Samsung Galaxy S2, LG Optimus 2x, HTC Sensation, etc.
- Gama extra alta: Samsung Galaxy S3, Sony Xperia S, etc.

Cada gama té les seves pròpies característiques. Així doncs els dispositius de gama baixa tenen unes prestacions més simples que els de gama mitjana, i així successivament. Les resolucions per cada gama les podem veure en la següent taula.

	Low density (120), <i>ldpi</i>	Medium density (160), <i>mdpi</i>	High density (240), <i>hdpi</i>	Extra high density (320), <i>xhdpi</i>
<i>Small screen</i>	QVGA (240x320)		480x640	
<i>Normal screen</i>	WQVGA400 (240x400) WQVGA432 (240x432)	HVGA (320x480)	WVGA800 (480x800) WVGA854 (480x854) 600x1024	640x960
<i>Large screen</i>	WVGA800** (480x800) WVGA854** (480x854)	WVGA800* (480x800) WVGA854* (480x854) 600x1024		
<i>Extra Large screen</i>	1024x600	WXGA (1280x800) [†] 1024x768 1280x768	1536x1152 1920x1152 1920x1200	2048x1536 2560x1536 2560x1600

Taula 3 - Resolucions i densitats

Una altra part important en el disseny es la dels elements de transició de pantalla a pantalla. Android té moltes llibreries implementades que proporcionen una funcionalitat òptima i a la vegada elegant. Quant als menús de l'aplicació com les transicions de pantalles s'han utilitzat un parell de llibreries. En primer lloc, "ActionBarsherlock" ens proporciona una barra amb la que es pot interactuar per fer una transició a diferents opcions o per tornar a la pantalla anterior, a més

a més, està dotada d'un disseny elegant. En segon lloc la utilització de Fragments, una tècnica molt utilitzada per a tauletes que ens permet fraccionar una pantalla en diferents parts, això ens permet adaptar dissenys de mòbil a tauletes i viceversa, fa el efecte en mòbils com si tinguéssim dues pantalles però en realitat hi ha una pantalla però fraccionada en dos. Amb aquesta tècnica per passar de Fragment a Fragment s'ha utilitzat un "Pager" que ens permet passar d'un Fragment a l'altra pulsant a la pantalla i movent el dit cap a un costat o un altre.

3.2 Disseny de la base de dades

En una aplicació que tracta dades es molt important l'ús d'una base de dades externa. D'aquesta manera es garanteix que no s'ha de modificar res del codi de l'aplicació quan hi han canvis en els continguts. Llavors, s'ha de saber quines són les dades crítiques i quines son les dades que poden canviar en un moment determinat.

Pel que fa a aquest projecte, no només s'han de tenir en compte les dades purament administratives (alumnes, professors, assignatures, ...) sinó també les dades per identificar els terminals, les relaciones entre alumnes, professors i assignatures, etc.

S'han creat 9 taules a la base de dades amb els seus camps corresponent. A continuació farem una breu descripció de cada una de les taules:

- **ASIGNATURAS:** Conté la informació rellevant de les assignatures com és el codi de l'assignatura, el nom i l'aula.

- AVISOS: Conté la informació necessària per emmagatzemar un avis de tipus notícia o material. A més et guarda de quina assignatura pertany l'avis i la seva data de registrament.
- CHAT: Conté les dades necessàries per la comunicació entre 2 alumnes, com és l'alumne emissor, l'alumne receptor i el missatge. A més guarda la data de registrament i la assignatura a la que pertany.
- MATRICULA: Conté la relació entre alumnes i assignatures.
- PROFESOR: Conté la informació necessària dels professors, com és el nom, cognoms, despatx i mail.
- PROFESORASIGNATURAS: Conté la relació entre professors i assignatures, a més guarda el paper del professor a la assignatura.
- TERMINALSAVISOS: Conté la relació entre els terminals i els avisos.
- TERMINALCHAT: Conté la relació entre els terminals i les comunicacions entre alumnes.
- USUARIOS: Conté les dades necessàries dels alumnes com és el nom, cognoms, NIU i password. A més conté l'identificador del terminal de cada alumne.

Per poder tenir una base de dades amb unes dades per poder fer proves s'ha omplert amb dades falses pel que fa als professors i als alumnes. Les dades dels alumnes i professors han sigut aleatòries. En canvi les dades de les assignatures han intentat semblar-se a les dades de la carrera d'Enginyeria Informàtica. Hi ha un total d'uns 80

alumnes, 80 assignatures i uns 81 professors. Cada assignatura té uns 15 alumnes i 1 professor excepte una que en té 2 professors.

3.3 Implementació de la interfície

Un dels avantatges de la programació en Android és la manera en com se estructura el codi. A continuació veurem les parts fonamentals del codi.

3.3.1 EL MANIFEST ANDROID

Aquest és el fitxer principal de qualsevol aplicació en Android, escrit en XML i es necessita per a que es pugui executar l'aplicació al sistema operatiu. A continuació veurem el Manifest del Campus Virtual UAB en versió reduïda i explicarem les parts més importants.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.uab.cvuab"
    android:versionCode="1"
    android:versionName="1.0">

    <uses-sdk
        android:minSdkVersion="9"
        android:targetSdkVersion="15" />

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.VIBRATE" /> <!-- Permís vibració.
    Utilitzat a les notificacions -->
    <permission
        android:name="com.androidbegin.gcmtutorial.permission.C2D_MESSAGE"
        android:protectionLevel="signature" />
    <uses-permission
        android:name="com.androidbegin.gcmtutorial.permission.C2D_MESSAGE" />
    <uses-permission android:name="com.google.android.c2dm.permission.RECEIVE" />
    <uses-permission android:name="android.permission.WAKE_LOCK" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <application
        android:allowBackup="true"
```

```

        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.uab.cvuab.SplashActivity"
            android:label="@string/app_name"
            android:theme="@android:style/Theme.Black.NoTitleBar">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name="com.uab.cvuab.LoginActivity"
            android:label="@string/app_name"
            android:windowSoftInputMode="adjustPan" >

        </activity>
        <activity
            android:name="com.uab.cvuab.HomeActivity"
            android:label="@string/app_name"
            android:windowSoftInputMode="stateHidden|adjustResize" >

        </activity>
        .
        .
        .
        <service android:name=".GCMIntentService" />
        <receiver
            android:name="com.google.android.gms.GCMBroadcastReceiver"
            android:permission="com.google.android.c2dm.permission.SEND"
            >
            <intent-filter>
                <action android:name="com.google.android.c2dm.intent.RECEIVE" />
                <action android:name="com.google.android.c2dm.intent.REGISTRATION" />

                <category android:name="com.androidbegin.gcmtutorial" />
            </intent-filter>
        </receiver>
    </application>

</manifest>

```

- **manifest:** és el TAG principal que engloba tot el fitxer. Es defineix el número de versió de l'aplicació.
- **application:** es defineix el tema per defecte, el icona i el nom de l'aplicació.
- **uses-permission:** defineixen els permisos de l'aplicació.

- **activity:** cada pantalla que apareix a l'aplicació s'ha de declarar amb aquest tag i definir l'atribut name que fa referència al fitxer que conté el codi que s'executa quan apareix la pantalla al dispositiu.
- **service:** és semblant al activity però la diferència és que fa referència a un servei remot.
- **receiver:** en aquest cas es declara un broadcast receiver que consisteix en una espècie de receptor d'events que produeix el dispositiu mòbil.

3.3.2 ORGANITZACIÓ DEL CODI

Una part important de la programació en Android i que ens facilita molt la feina és la seva forma d'organitzar el codi. El codi s'estructura per carpetes. A continuació veurem la seva estructuració per carpetes.

- **src:** en aquesta carpeta aniran tots els arxius JAVA que fonamentalment són els arxius del motor de l'aplicació.
- **gen:** conté una sèrie d'elements de codi generats automàticament al compilar l'aplicació. Aquest fitxers estan dirigits al control de recursos de l'aplicació.
- **bin:** en aquesta carpeta aniran tots els arxius executables de l'aplicació que es generen automàticament al compilar l'aplicació.
- **libs:** aquí aniran totes les llibreries incloses en el projecte.
- **res:** dins d'aquesta carpeta tenim tot allò que té a veure amb la part gràfica de l'aplicació. Dins podrem trobar les

carpetes drawables que consisteixen en guardar les imatges o icones de l'aplicació, la carpeta layout que és on es defineixen les pantalles de l'aplicació escrit en XML, la carpeta menú que és on es defineixen els menús de l'aplicació i la carpeta values que es on es defineixen els textos, colors i estils de l'aplicació.

3.3.3 FLUX DEL PROGRAMA

A continuació veurem un gràfic que es mostra quin es el recorregut de l'aplicació.

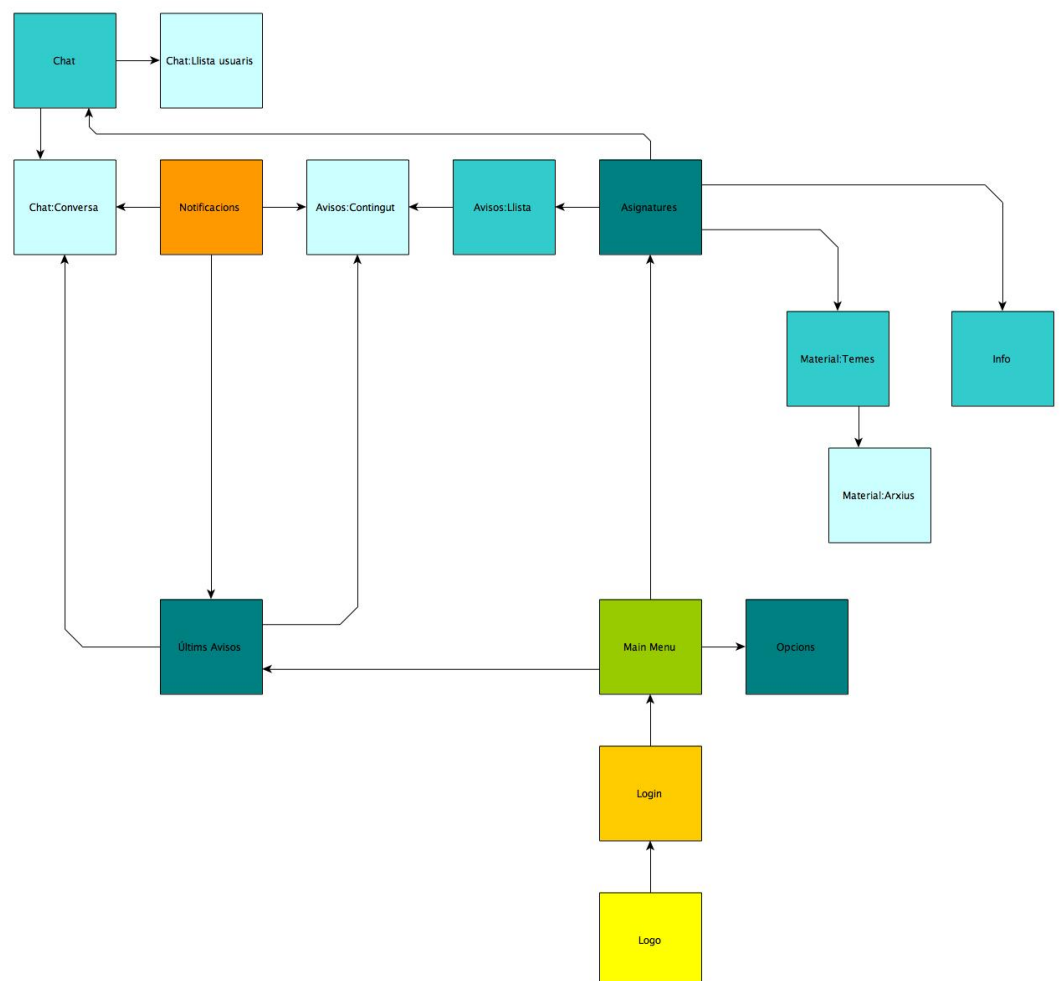


Figura 3 - Flow del programa

Pel que fa al fluxe del programa, l'aplicació s'inicia amb el logo de la UAB com a introducció al programa i seguidament si és el primer cop que hi entres, et demanarà el NIU i el password de l'alumne per entrar. Aquest pas només es farà el primer cop que s'obri l'aplicació gracies a les SharedPreference que et permeten guardar variables globals de l'aplicació. Cada cop que hi entrem es comprova que no siguin buides. Un cop que es comprova el NIU i password es comença a carregar les dades. Un cop les dades s'han carregat, com ja s'ha esmentat abans, s'ha utilitzat la tècnica dels Fragments i el primer que ens trobem son 2 fragments. En el primer Fragment tenim un llistat d'assignatures i en el segon un llistat d'avisos. Cada llistat esta format per un ListView que ens permet crear llistes dins una vista. A la barra de dalt (ActionBarSherlock) tenim una icona que ens indica la acció per anar a les opcions del programa. Per passar d'un fragment a l'altre s'ha utilitzar un Pager perquè sigui més intuïtiu i més modern, encara que també es pot fer clickant a un dels Tabs que permeten amagar informació i només mostrar la seleccionada.



Asignaturas	Últimos Avisos
Teoría de Autómatas	Josep Bacardit Sáez QC-1001
Lógica Computacional	Charles Green QC-1006
Inteligencia Artificial 1	Diego Ruiz Hood QC-1011
Álgebra Lineal	Francesc Tost Rodríguez QC-1017
Estructura de Computadores 1	Carles Vilalta Aguilar QC-1022
Estructura de Computadores 2	Montse Lapiedra Ramos QC-1026
Inteligencia Artificial 2	Aitor Robiralta Moreno QC-1032
Compiladores 2	Ricardo Craviotto Vilchez QC-1037

Figura 4 - Captura asignaturas



Asignaturas	Últimos Avisos
Teoría de Autómatas	Material Ya teneis en material los apuntes del tema 4. 27/8/2013
Inteligencia Artificial 1	Noticia Las clases no empiezan hasta el 25 de setiembre. 27/8/2013

Figura 5 - Captura avisos

Si cliquem a una de les assignatures o en algun dels avisos passarem a la pantalla corresponent a la assignatura seleccionada. Aquesta pantalla té 2 Fragments també, el primer ens mostra el material de l'assignatura organitzat per carpetes fent ús de la tècnica ExpandableList, que ens permet crear llistes que es poden expandir clicant a un dels elements, i el segon ens mostra els avisos d'aquella assignatura de la mateixa manera que en la pantalla anterior. També tenim 2 icones a la barra de dalt, una ens porta a la missatgeria i l'altra a la informació de l'assignatura.

Si cliquem al icona de la missatgeria anirem a la pantalla dels missatges rebuts i enviats. Aquesta pantalla consta de 2 Fragments. El primer ens mostra en un ListView els missatges

rebut i el segon ens mostra els missatges enviats. A la barra de dalt tenim un icona que ens permet anar a la pantalla on surt el llistat d'alumnes de l'assignatura per poder enviar un missatge nou.



Figura 6 - Captura mtge rebuts



Figura 7 - Captura mtge enviats

3.3.4 ESTRUCTURA DE DADES

Quan es treballa amb un projecte gran i hi ha tanta quantitat d'informació no es pot treballar amb els tipus de dades per defecte ja que ens dificultaria molt la feina. Es per això que existeixen les estructures de dades i en el cas del JAVA les classes dels objectes. Per poder treballar amb les dades que rebem de la base de dades s'han hagut de crear nous tipus de dades. A continuació esmentarem quin són aquest tipus de dades i farem una petita descripció de la seva funcionalitat.

- **Asignatura:** és el tipus de dades més important de l'aplicació, ja que, centralitza gairebé tota la informació de l'aplicació. En aquest tipus emmagatzemem el codi de l'assignatura, el nom, una llista de professors, l'aula, una llista d'alumnes i una llista d'avisos.
- **Alumno:** en aquest tipus de dades emmagatzemem la informació essencial del alumne. Guardem el NIU, password, nom i cognoms.
- **Profesor:** en aquest tipus de dades emmagatzemem la informació essencial del professor. Guardem la id del professor, el nom i cognoms, despatx i mail.
- **Profasig:** en aquest tipus de dades guardem un tipus de dades Profesor i guardem el paper del professor (Teoria o Pràctiques) i si es titular de la assignatura o no. Aquest tipus de dades està inclòs al tipus de dades Asignatura.
- **Matricula:** en aquest tipus de dades guardem un tipus de dades Alumno i la nota final del alumne. Aquest tipus de dades està inclòs al tipus de dades Asignatura.

- Avisos: en aquest tipus de dades guardem la informació dels avisos. S'emmagatzema la id del avis, l'assignatura, el títol del avís, la data de registrament, el tipus d'avis i el missatge del avís.
- Conversa: en aquest tipus de dades guardem la informació dels missatges entre alumnes. Es guarda la assignatura a la que pertany, l'alumne qui rep el missatge, el missatge i la data d'enviament.

Tots aquest tipus de dades tenen una cosa en comú. Totes hereten de la classe Parcelable. Com que la majoria de l'informació de la base de dades es carrega al inici, s'han de poder passar les dades de pantalla a pantalla. Es podria anar passant una a una però això es molt molest. Gràcies a la classe Parcelable, ens permet passar un objecte entre pantalles i fa que la programació sigui mes efectiva i a la vegada més fàcil de llegir.

3.4 Accessos a la base de dades

Hi han dos maneres per dotar de contingut a una aplicació. D'una banda, es pot crear una base de dades interna a la aplicació. Aquesta opció és poc recomanable si les dades de l'aplicació contínuament poden anar canviant, ja que, això ens faria que cada cop que hi hagués un canvi, s'hauria d'actualitzar l'aplicació i això no es gaire seriós, ja que, dóna la sensació de ser una aplicació mal acabada. D'altra banda, es pot utilitzar una base de dades externa a la aplicació. Aquest es el cas òptim per aplicacions que contínuament

poden anar canviant de contingut. Es per aquest motiu que aquest projecte utilitza una base de dades externa.

Llavors, per poder crear una base de dades externa és necessari tenir un servidor. El més ideal és tenir un servidor amb un IP fixa, d'aquesta manera, sabem sempre on tenim la base de dades i a nivell de programació és molt més senzill. Quant a aquest projecte, no s'ha pogut adquirir cap servidor amb una IP fixa, la qual cosa, m'ha fet que escollís un servidor local. Al treballar amb un MAC, per poder muntar el servidor, vaig utilitzar el software MAMP.

MAMP ens permet muntar un servidor de forma local sense gairebé configurar res. A més a més, s'ha utilitzat la eina PhpMyAdmin per poder crear i gestionar la base de dades.

Podem diferenciar dues parts en els accessos a la base de dades. D'una banda, tenim la part de l'aplicació i d'altra banda, la part del servidor.

3.4.1 LA PART DE L'APLICACIÓ

Pel que fa a la part de l'aplicació s'ha centralitzat tots els accessos en una classe. Aquesta classe s'anomena CVWebservices. Per poder connectar-se al servidor es necessari que se li especifiqui la IP on està ubicat el servidor. Per poder fer els accessos a la base de dades, s'ha desenvolupat la classe CVWebservices, la qual engloba un conjunt de mètodes que segons quina informació es vol rebre s'executa un o altre.

A continuació podem veure un exemple d'un dels mètodes de la classe. En aquest exemple el que es pretén es obtenir els alumnes d'una assignatura.

```
public ArrayList<Alumno> parseAlumnosResponse(String asignatura,String niu)
throws IOException {

    ArrayList<Alumno> alumnosList = new ArrayList<Alumno>();

try {
    StringBuffer completeUrl = new StringBuffer(url + "alumnos.php");

    completeUrl.append("?");

    completeUrl.append("asignatura="+asignatura);
    completeUrl.append("&");
    completeUrl.append("niu="+niu);
    completeUrl.append("&");
    completeUrl.append("format=json");

    String dataAsString = getDataAsString(completeUrl);

    JSONObject jsonResponse = new JSONObject(dataAsString);

    JSONArray alumnosArray= jsonResponse.getJSONArray("alumnos");

    for(int i=0; i<alumnosArray.length(); i++) {
        JSONObject obj = alumnosArray.getJSONObject(i).getJSONObject("alumno");
        Alumno alumno = new Alumno();
        alumno.setNiu(obj.getString("NIU"));
        alumno.setPassword(obj.getString("Password"));
        alumno.setNombre(obj.getString("Nombre"));
        alumno.setApellidos(obj.getString("Apellidos"));
        alumnosList.add(alumno);
    }

} catch (JSONException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
} catch (NullPointerException e) {
    // TODO: handle exception
    e.printStackTrace();
}

return alumnosList;
}
```

En l'exemple, se li passa com a paràmetres el codi de l'assignatura i el niu de l'alumne que està utilitzant l'aplicació per quant s'obtinguin els alumnes no hi surti ell mateix. Seguidament se l'indica quin és el PHP que ha d'executar amb els seus paràmetres per poder fer la petició al servidor. Un cop feta la petició, el següent pas es rebre les dades i guardar-les en alguna estructura. Aquesta estructura, en aquest cas es el tipus de dades Alumne i guardem tots els alumnes en un array d'aquest tipus de dades. Finalment, retornem l'array amb tots els alumnes obtinguts.

3.4.2 LA PART DEL SERVIDOR

Pel que fa a la part del servidor, com s'ha esmentat anteriorment, el que fem són peticions en PHP. En aquest cas, tenim per cada petició al servidor, un fitxer PHP diferent que segons que se li demani des de l'aplicació s'executarà un o altre.

A continuació veiem un exemple d'un dels fitxers PHP. Per seguir amb l'exemple anterior, el fitxer PHP que veurem fa la petició al servidor per obtenir tots els alumnes d'una assignatura.

```
<?php
/* require the user as the parameter */

$assignatura = $_GET['assignatura'];
$niu = $_GET['niu'];

/* soak in the passed variable or set our own */
$number_of_posts = isset($_GET['num']) ? intval($_GET['num']) : 8; //7 is the
default
$number_of_posts = !isset($_GET['all']) ? $number_of_posts : FALSE;
$format = strtolower($_GET['format']) == 'json' ? 'json' : 'xml'; //xml is the
default
```

```

/* connect to the db */
$link = mysql_connect("localhost","root","root") or die('Cannot connect to the DB');
mysql_select_db("cv",$link) or die('Cannot select the DB');
mysql_set_charset('utf8',$link);

/* grab the posts from the db */
$query = "SELECT NIU, Password, Nombre, Apellidos FROM USUARIOS WHERE NIU IN
(SELECT NIU FROM MATRICULA WHERE Id_Asignatura='".$$_asignatura.'" AND
NIU<>".$_niu."")";

$result = mysql_query($query,$link) or die(mysql_error());

/* create one master array of the records */
$posts = array();
$rows = 0;
if($rows = mysql_num_rows($result)) {
    while($post = mysql_fetch_assoc($result)) {
        $posts[] = array('alumno'=>$post);
    }
}

/* output in necessary format */
if($format == 'json') {
    header('Content-type: application/json; charset=UTF-8');
    echo json_encode(array('alumnos'=>$posts));
}
else {
    header('Content-type: text/xml; charset=UTF-8');
    echo 'alumnos';
    foreach($posts as $index => $post) {
        if(is_array($post)) {
            foreach($post as $key => $value) {
                echo '<',$key,'>';
                if(is_array($value)) {
                    foreach($value as $tag => $val) {
                        echo '<',$tag,'>',htmlentities($val,
                        ENT_COMPAT, "UTF-8"),'</',$tag,'>';
                        //echo '<',$tag,'>',$val,'</',$tag,'>';
                    }
                }
                echo '</',$key,'>';
            }
        }
    }
    echo '</alumnos';
}
/* disconnect from the db */
@mysql_close($link);

?>

```


En l'exemple, primer de tot obtenim els paràmetres que rebem de l'aplicació i a continuació ens connectem a la base de dades i fem la consulta per rebre els alumnes d'una assignatura. Un cop feta la consulta, passem al pas de rebre les dades. Aquest fitxer PHP està preparat per poder mostrar les dades en XML o en JSON. En el nostre cas utilitzem l'estandard JSON, ja que, ens facilita el treball a l'hora de parsejar les dades. Un cop feta la consulta, segons si volem mostrar les dades en XML o en JSON, creem la estructura perquè l'aplicació pugui agafar les dades.

3.5 Notificacions

Aquesta és la tasca més important del projecte. De que serviria una aplicació del Campus Virtual si no hi haguessin notificacions. Seria pràcticament igual que via web. La gràcia d'aquesta aplicació es que quan hi hagi alguna noticia o qualsevol tipus d'avis, que et pugui notificar. D'aquesta manera, sempre t'adonaràs quan hi hagin novetats.

Les notificacions en Android es realitzen amb el servei de Google GCM (Google Cloud Messaging). És un servei que et permet enviar missatges des de el teu servidor als usuaris. Per poder utilitzar aquest servei s'ha de registrar l'aplicació a Google i d'aquesta manera s'obté una "API Key" que servirà com identificador de l'aplicació amb el servei.

3.5.1 IMPLEMENTACIÓ GCM CLIENT

El primer que s'ha de fer a la part del usuari, és a dir, a la part de l'aplicació, és modificar el fitxer Manifest. S'han de afegir uns permisos. Aquest permisos són els següents.

- **com.google.android.c2dm.permission.RECEIVE:** Aquest permís permet a les aplicacions enviar i rebre missatges.
- **android.permission.INTERNET:** Ens permet poder connectar a internet les aplicacions i per tant, en aquest cas ens permet enviar el identificador del terminal.
- **android.permission.GET_ACCOUNTS:** Per poder utilitzar aquest servei es necessari tenir una compte de Google, aquest servei accedeix al nostre compte.
- **android.permission.WAKE_LOCK:** Aquest permís en assegura que si l'aplicació està en segon pla que pugui estar pendent per rebre missatges.
- **Application Package + .permission.C2D_MESSAGE:** Dóna el permís per rebre missatges.
- **receiver de com.google.android.c2dm.intent.RECEIVE:** Ve definit en la libreria GCM i necessita d'un altre permís (com.google.andrpod.c2dm.SEND) i és el receptor dels missatges.
- **Service:** és l'encarregat de crear la notificació a dalt de la barra. Es pot incloure un IntentService que permet a l'usuari obrir l'aplicació quan clickem a sobre d'una de les notificacions.
- Per utilitzar aquest servei es necessari que el sdk d'android sigui com a mínim la versió 8.

El següent pas, consisteix en la implementació del Service. Aquest Service rebrà el nom de GCMIntentService i ens permetrà registrar el terminal i rebre els missatges per crear les notificacions. Un cop implementat el Service em de dir-li el moment en el que volem registrar el terminal, en el nostre cas, és al moment d'entrar a l'aplicació i per registrar-lo el que fem és guardar a la base de dades l'identificador del terminal associat al niu del alumne. Indicant-li el moment, GCMIntentService rep un event i executa el mètode onRegistered. Aquest mètode rep com a parametre el identificador del terminal. Quan es rep una notificació, GCMIntentService rep un event i executa el mètode onMessage. Aquest mètode rep com a paràmetre el contigut del missatge.

3.5.2 IMPLEMENTACIO GCM SERVIDOR

El servidor és qui s'encarrega de enviar els missatges. A continuació veiem la seqüència general per enviar els missatges.

1. El servidor envia un missatge als servidors de GCM.
2. Google fica els missatges en una cua en el cas que el terminal no estigui disponible.
3. Quan el terminal està disponible, Google envia els missatges.
4. En el terminal es captura el missatge encara que la aplicació estigui en segon pla.

5. L'aplicació processa els missatges.

El servidor envia els missatges executant uns fitxers PHP. En aquest fitxers el primer que es fa es fer una consulta a la base de dades per si s'ha de enviar algun tipus d'avis. En el cas que hi hagi algun avís pendent, el servidor captura els avisos i els terminals dels alumnes que han de rebre els avisos. Després crea el missatge amb els identificadors dels terminals i els envia al servidors de GCM. Llavors, Google ja se encarrega de que arribin als terminals desitjats. Un cop enviat, canviem a la base de dades l'estat dels avisos com a enviats per tal que no es torni a enviar de nou el proper cop.

La execució dels fitxers es fa de forma manual. La idea, com s'explicarà en un dels annexos, és que aquesta tasca estigui automatitzada amb un CRON.

3.6 Proves

Aquesta tasca es la més llarga de tot el projecte, ja que, hi es present durant gairebé tota la durada del projecte. Les proves en un projecte són molt importants, ja que, es detectem els errors a nivells de execució i a nivells gràfics, visualitzem noves idees, etc.

En l'únic moment en que no hi han proves són en les primeres fases del projecte quan encara no hi ha codi desenvolupat i només se està en fase de disseny. En el moment que comença la programació, contínuament se estan fent proves.

3.6.1 TERMINALS

No s'han pogut fer proves amb gaires terminals. Els terminals utilitzats es descriuen a continuació.

SONY XPERIA S



Figura 8 - Sony Xperia S

El **Sony Xperia S** és un smartphone Android amb una pantalla HD de 4.3 polsades, amb una resolució de 720 x 1280 pixels y un processador dual-core a 1.5GHz. A més a

més, té una càmera amb sensor Exmor-R de 12 megapixels, així com també una càmera frontal. Quant a la memòria, està disponible en versions de 16GB y 32GB, encara que no disposa de ranura de expansió de memòria microSD. A més a més, Actualment el seu sistema operatiu té la 4.2 d'Android (Jelly Bean).

SAMSUNG GALAXY NOTE 2



Figura 9 - Samsung Galaxy Note 2

El Samsung GALAXY Note II és el successor del Galaxy Note original. El Galaxy Note II porta la pantalla Super AMOLED HD a 5.5 polsades, i agrega un processador Exynos quad-core a 1.6GHz,

càmera posterior de 8 megapixels, càmera frontal de 1.9 megapixels y funcionalitat S-Pen extesa. El seu sistema operatiu es el 4.2 d'Android (Jelly Bean).

EMULADOR ANDROID

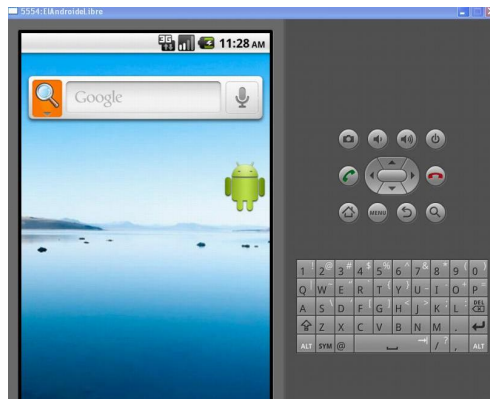


Figura 10 - Emulador Android 1

S'ha utilitzat el emulador d'Android que ens proporciona el ADT plugin d'Eclipse. Les característiques configurades del emulador simulen a un Nexus S amb

una pantalla de 4 posades i una resolució de 480x800. De memòria ram amb 343 mb i amb 512 mb de memòria interna. El sistema operatiu simulat és el 2.3.3 (Gingerbread).

El terminal més utilitzat per fer les proves ha sigut el Sony Xperia S. S'han fet proves de elements gràfics, de connexions a bases de dades, de execució de tasques i de notifikacions.

Pel que fa al Samsung Galaxy Note 2, s'ha utilitzat juntament amb el Xperia S per fer les proves de comunicació entre alumnes. Les proves amb l'emulador d'Android van ser al inici de la programació del projecte per veure les primeres captures gràfiques de l'aplicació.

3.6.2 ANDROID LOG

El log d'Android es una de les eines més importants per el desenvolupament d'aplicacions d'Android. Gràcies al plugin ADT d'Eclipse, podem executar les aplicacions amb el nostre terminal i alhora veure que està passant per dins del terminal. Amb el log d'Android podem veure si en un moment determinat la execució de l'aplicació passa per un lloc determinat del codi, podem imprimir el valor d'una variable en qualsevol moment, etc. També és un lloc per imprimir els missatges d'errors que ens dona JAVA. És una eina que ens ajuda al control de qualitat de l'aplicació.

A més a més, el SDK d'Android incorpora una classe que ens permet imprimir qualsevol missatge i a qualsevol del següents nivells.

- **v (String,String)** (Verbose)
- **d (String,String)** (Debug)
- **i (String,String)** (Information)
- **w (String,String)** (Warning)
- **e (String,String)** (Error)

Un exemple seria:

```
Log.i("MyActivity", "MyClass.getView() — get item number " + position);
```

4. Conclusions

La idea del projecte era poder crear una aplicació en Android per poder veure el Campus Virtual als dispositius mòbils sense haver d'utilitzar cap tipus de navegador. Per tant, no només es volia fer una eina que facilités l'ús del Campus Virtual, sinó que és volia millorar algunes mancances que té el Campus Virtual.

4.1 *Objectius assolits i millores*

Avui en dia, gairebé tothom porta un smartphone a sobre, es per això que moltes aplicacions fan ús de notificacions quan volen que els usuaris es vulguin assabentar d'alguna cosa. Una de les mancances del Campus Virtual és que si no entres al web no te n'assabentes del que hi ha i moltes vegades ens a ha passat que ens hem deixat coses enrere per no llegir les novetats. Amb aquest projecte, aquesta mancança s'ha resolt amb les notificacions. Quan hi ha alguna novetat, se l'hi notifica al usuari. Per gestionar les notificacions, en el cas de les notícies o avisos de nou material, les notificacions es creen via web i es guarden a la base de dades, després s'executa un PHP manualment per llançar les notificacions. D'altra banda, per notificar un missatge d'un usuari a un altre, els usuaris a través de l'aplicació generen els missatges i es guarden a la base de dades, després s'executa un PHP manualment que s'encarrega d'enviar les notificacions. Una de les possibles millores per aquest projecte seria poder automatitzar aquesta tasca, d'aquesta manera, quan es generi una notificació, cada 10 segons es vagi mirant si hi han noves notificacions a la base de dades i es llancin quan faci falta.

Cal dir que aquesta part del projecte ha sigut la més difícil per a mi. Jo ja havia fet alguns projectes pel meu compte en la programació per Android, però mai havia fet ús de notifikacions. És per això que ha sigut la feina més difícil, ja que, s'ha hagut de trobar informació i després poder-la aplicar amb les eines de desenvolupament que he utilitzat.

D'altra banda, es volia separar la part de l'interfície amb la part del contingut. Una aplicació que pugui variar el seu contingut contínuament no pot barrejar aquest dos conceptes. Es per això que s'ha utilitzat una base de dades externa a l'aplicació, d'aquesta manera ens evitem que cada cop que hi hagi un canvi en el contingut no s'hagi de actualitzar l'aplicació. Com a millora es podria fer un gestor de continguts per canviar el contingut i gestionar els avisos. Aquest gestor de continguts seria via web, ja que, es la manera més eficient i senzilla de fer i la més còmoda per la persona encarregada de fer els canvis.

Quant al tema dels missatges, la idea inicial es que no només es pogués enviar missatges entre alumnes, sinó també que es pogués enviar missatges als professors. Aquesta idea va quedar descartada per aquest projecte, ja que, s'hauria de crear una altra aplicació especial pel professorat per poder aprofitar els avantatges que tenen els dispositius mòbils. Per tant, com a millora s'hauria de crear aquesta aplicació per poder intercanviar missatges entre alumnes i professors.

A causa de no tenir més dispositius al meu abast, només s'ha pogut testejar en els dispositius ja descrits anteriorment en l'apartat de proves. M'hagués agradat fer proves amb tauletes, ja que, penso que pot ser més útils en tauletes, ja que les pantalles son més grans i es pot llegir la informació amb més claredat.

4.2 A on es vol arribar?

Com ja s'ha comentat a la introducció, aquesta aplicació no s'ha de quedar en un simple projecte. La meua intenció des d'abans de la realització del projecte es que fos una eina pels futurs estudiants de la UAB.

En primer lloc, s'hauria de completar l'aplicació amb noves funcionalitats que ja estan incloses al Campus Virtual via web. Aquestes funcionalitats son el gestor i inscripcions de grups, fòrum, lliurament de treballs, etc.

En segon lloc, el que es pretén es centralitzar en una aplicació tot el que té a veure amb la universitat. La idea es que totes les assignatures utilitzin aquesta eina, que no hi hagi una assignatura que proporcioni el seu material per un cantó i una altra assignatura per un altre cantó. Que les inscripcions dels grups de pràctiques es facin mitjançant aquesta eina. Que a través de la missatgeria d'aquesta aplicació s'elimini l'ús del correu electrònic. Per tant, que aquesta aplicació sigui l'enllaç entre les assignatures i tot el que envolta a la UAB amb els estudiants.

I per últim, aquest projecte està fet per Android però no ens em d'oblidar que existeixen més sistemes operatius per dispositius mòbils. La idea es que es pugui fer servir per a IOS que juntament amb Android són els dos sistemes que engloben gairebé tota la telefonia mòbil i tauletes del mercat.

5. Bibliografia

- Desenvolupament
 - Api d'Android: <http://developer.android.com>
 - StackOverflow: <http://stackoverflow.com>
 - ActionBarSherlock: <http://actionbarsherlock.com>
 - Notificacions GCM: <http://www.sgoliver.net/blog/?p=4126>
- Llibres
 - Meier, Reto. Professional Android Application Development
2. Ed. Wiley Publishing, 2010
 - Murphy, Mark L. Beginning Android. Ed. Apress, 2009

Memòria del Projecte realitzat per

Josep Bacardit Sáez

Bellaterra, 16 de setembre de 2013

El projecte Campus Virtual UAB pretén ser una eina pels futurs estudiants de la UAB, destinat per la plataforma Android, amb l'objectiu de facilitar l'estudi dels alumnes. En aquesta memòria es descriu el desenvolupament del projecte indicant quines han estat les parts més importants per a la realització de l'aplicació.

El proyecto Campus Virtual UAB pretende ser una herramienta para los futuros estudiantes de la UAB, destinado para la plataforma Android, con el objetivo de facilitar el estudio de los alumnos. En esta memoria se describe el desarrollo del proyecto indicando que partes han sido más importantes para la realización de l'aplicación.

The Campus Virtual UAB project pretend to be a tool for the future students of UAB, destinated for the platform Android. With the objective to facilitate the study of the students. In this thesis, it's described the development of the projects indicating which parts have been more important for the realization of the application.